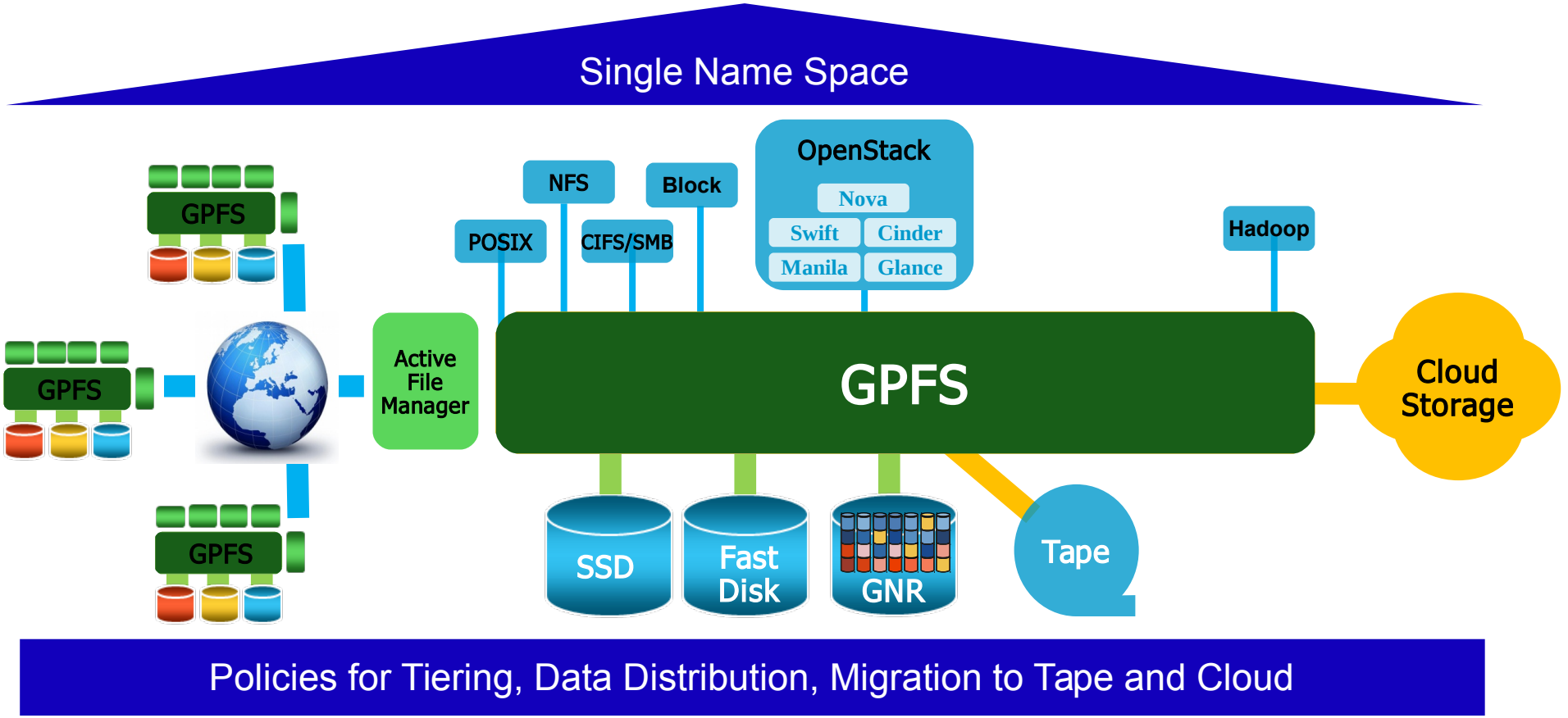
The Samba performance challenge :-)

Sven Oehme – oehmes@us.ibm.com

IBM Software defined file storage – the big picture



Bandwidth – one way to look at the problem

Gigabit	100 MB/sec
10 Gbit	1,000 MB/sec
40 Gbit QDR IB	4,000 MB/sec
56 Gbit FDR IB	5,600 MB/sec
NL SAS drive 100% Seq Read/Write	100 MB/sec
NL SAS drive 100% 2MB Random Read/Write	75 MB/sec
SSD 100% Seq Read	400 MB/sec
SSD 100% Seq Write	300 MB/sec

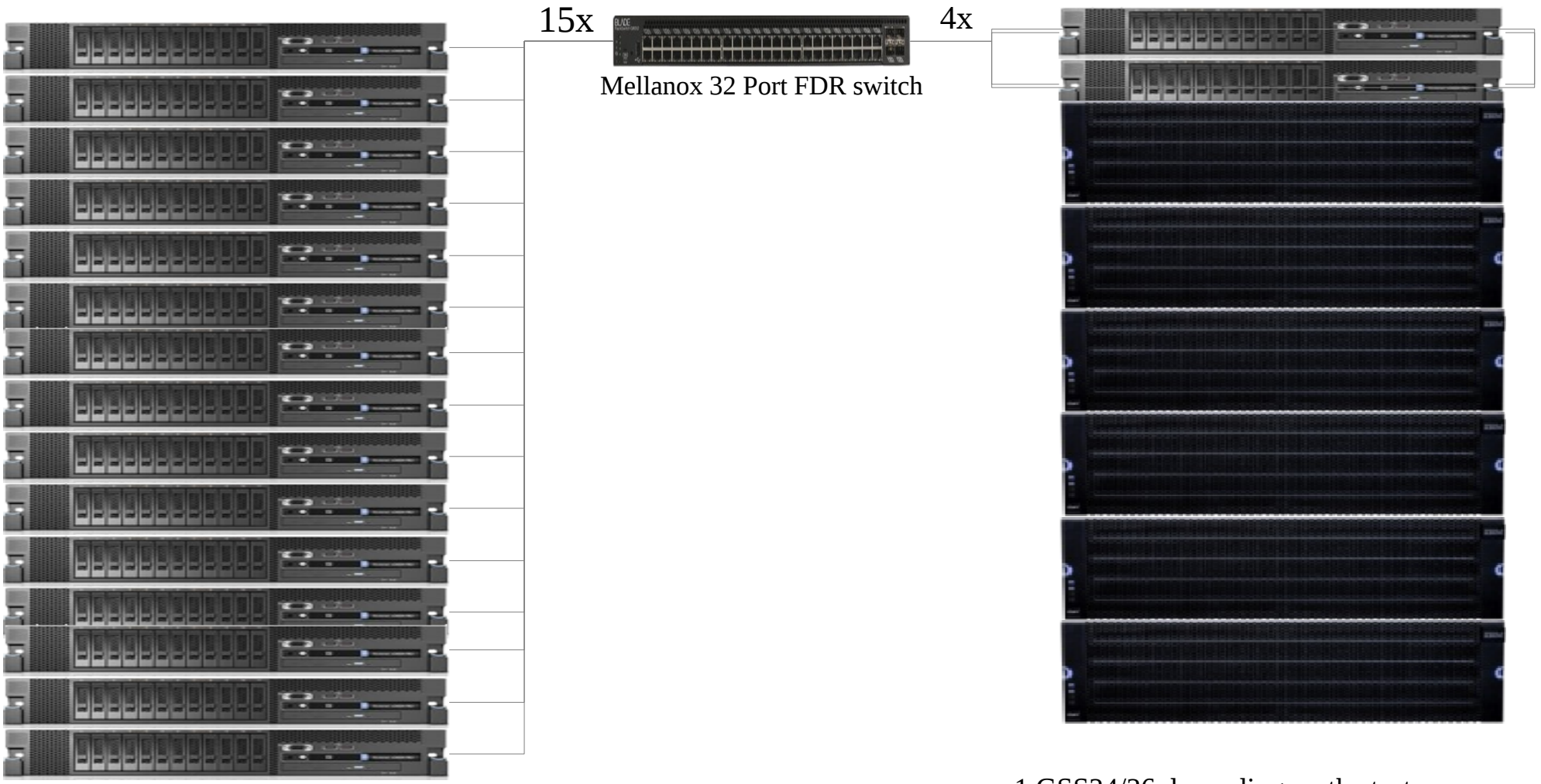
*These numbers are rounded and don't claim to be 100% accurate

IOPS numbers – another

NL SAS drive 100% 4k Random iops	100
10k SAS drive 100% 4k Random iops	200
SSD 100% 4k random reads	50,000
SSD 100% 4k random writes	20,000

*These numbers are rounded and don't claim to be 100% accurate

High Performance Storage Benchmark Setup



15 x3550-M3 Server each with
16 GB of Memory (6 gb Pagepool)
1 FDR Port
1 x 6 core CPU

1 GSS24/26 depending on the test.
2 FDR Ports connected per Server
GPFS 3.5.0.7 GA code level

Benchmark results across multiple nodes

```
ior -i 2 -p -d 10 -w -r -e -t 16m -b 32G -o /ibm/fs2-16m/shared/ior//iorfile

-i N  repetitions -- number of repetitions of test
-d N  interTestDelay -- delay between reps in seconds
-w    writeFile -- write file
-r    readFile -- read existing file
-e    fsync -- perform fsync upon POSIX write close
-t N  transferSize -- size of transfer in bytes (e.g.: 8, 4k, 2m, 1g)
-b N  blockSize -- contiguous bytes to write per task  (e.g.: 8, 4k, 2m, 1g)
-o S  testFile -- full name for test
```

Operation	4m	16m
GSS26-write (MB/sec)	11302.19	14970.40
GSS26-read (MB/sec)	13915.36	15193.71
GSS24-write (MB/sec)	7799.26	11148.37
GSS24-read (MB/sec)	9515.66	13875.70

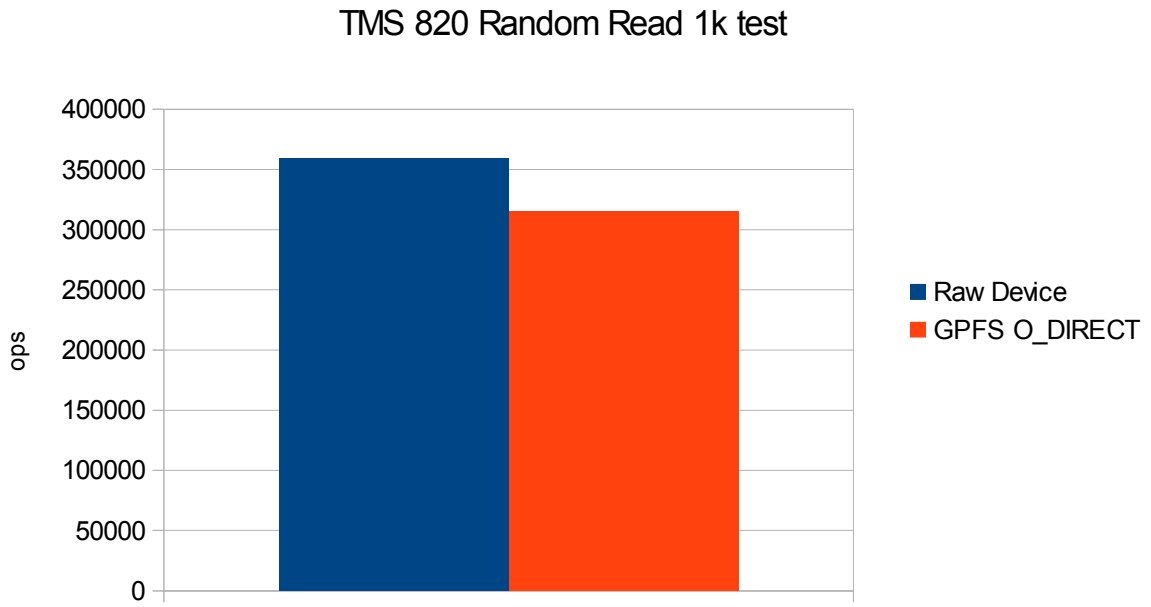
Sequential results from a single node

Creating a single 10 Gbyte File from one Client using a GEN-2 FDR IB card

```
# /usr/local/bin/gpfsperf create seq -n 10G -r 8m /ibm/fs2-8m/test-10g-write
/usr/local/bin/gpfsperf create seq /ibm/fs2-8m/test-10g-write
  recSize 8M nBytes 10G fileSize 10G
  nProcesses 1 nThreadsPerProcess 1
  file cache flushed before test
  not using data shipping
  not using direct I/O
  offsets accessed will cycle through the same file segment
  not using shared memory buffer
  not releasing byte-range token after open
  no fsync at end of test
    Data rate was 3268199.54 Kbytes/sec, iops was 398.95, thread utilization 0.984
    Record size: 8388608 bytes, 10737418240 bytes to transfer, 10737418240 bytes transferred
    CPU utilization: user 3.68%, sys 3.87%, idle 92.45%, wait 0.00%
```

Rate is limited by GEN-2 PCI card, GEN-3 cards deliver **5.6 GB/sec**

RamSan TM-820 single client random read test

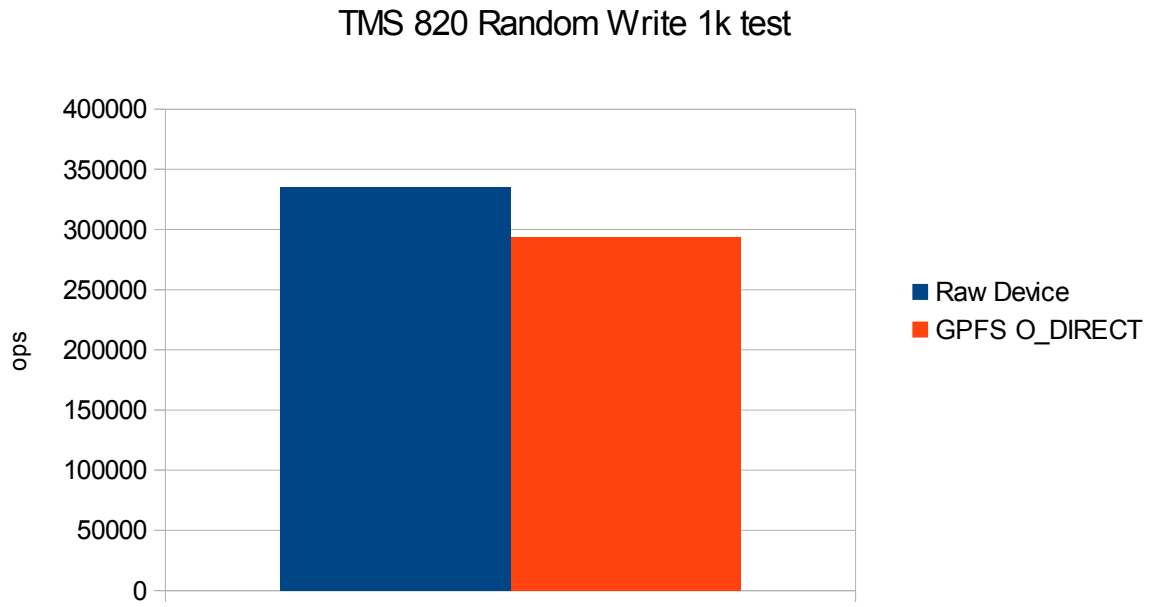


This Number of Operations was generated on just 2 files opened with gpfsperf like

/usr/local/bin/gpfsperf read rand -r 1k -n 10g /ibm/tms-256k/file-$num -th 64 -dio

A word of caution : The achieved numbers depends on the right Client configuration and good Interconnect and can vary between environments. They should not be used in RFI's as commited numbers, rather to demonstrate the technical capabilities of the Product in good conditions

RamSan TM-820 single client random write test



This Number of Operations was generated on just 2 files opened with gpfsperf like

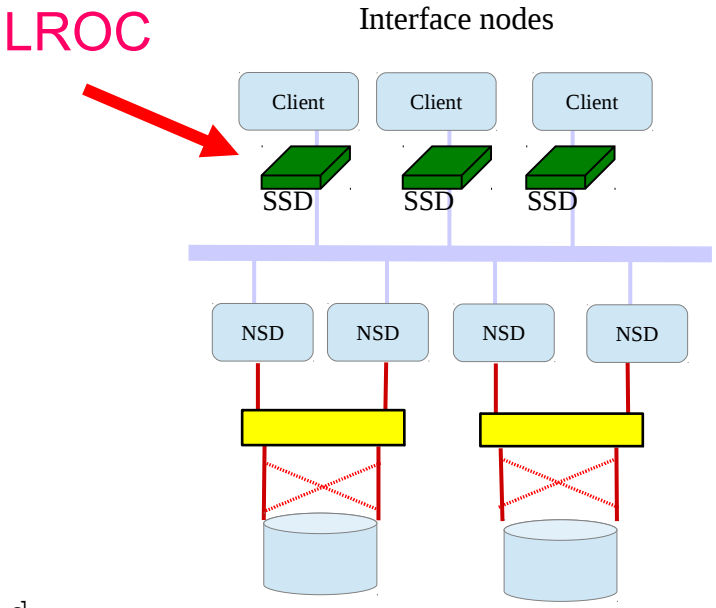
/usr/local/bin/gpfsperf write rand -r 1k -n 10g /ibm/tms-256k/file-$num -th 64 -dio

A word of caution : The achieved numbers depends on the right Client configuration and good Interconnect and can vary between environments. They should not be used in RFI's as commited numbers, rather to demonstrate the technical capabilities of the Product in good conditions

other industry trends relevant

GPFS Flash Local Read Only Cache (LROC)

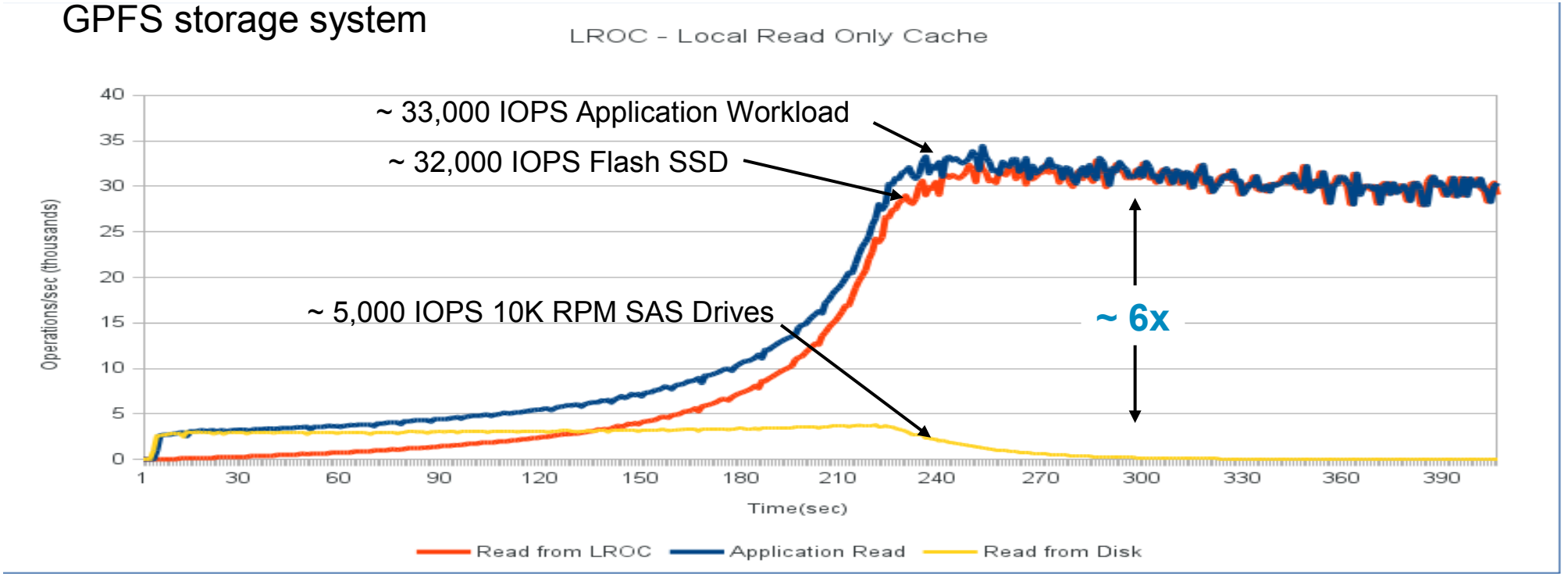
- Many NAS workloads benefit from large read cache
  - SPECsfs
  - VMWare and other virtualization
  - Database
- Augment the Interface Node DRAM cache with SSD
  - Used to cache:
    - Data
    - Inodes
    - Indirect blocks
  - Cache consistency insured by standard GPFS tokens
  - Assumes SSD device is unreliable, data is protected by checksum and verified on read
  - Provide low-latency access to file system metadata and data
- Implement with consumer flash for maximum Cache/$
  - Enabled by FLEA's LSA (Data is writte Sequential to Device, to eliminate wear leveling)
  - Reach small File performance leadership compared to other NAS Devices



Add 100's of GBs of SSD to each interface node

GPFS Flash Cache Example Speed Up

- Two consumer grade 200 GB SSDs cache a forty-eight 300 GB 10K SAS disk GPFS storage system



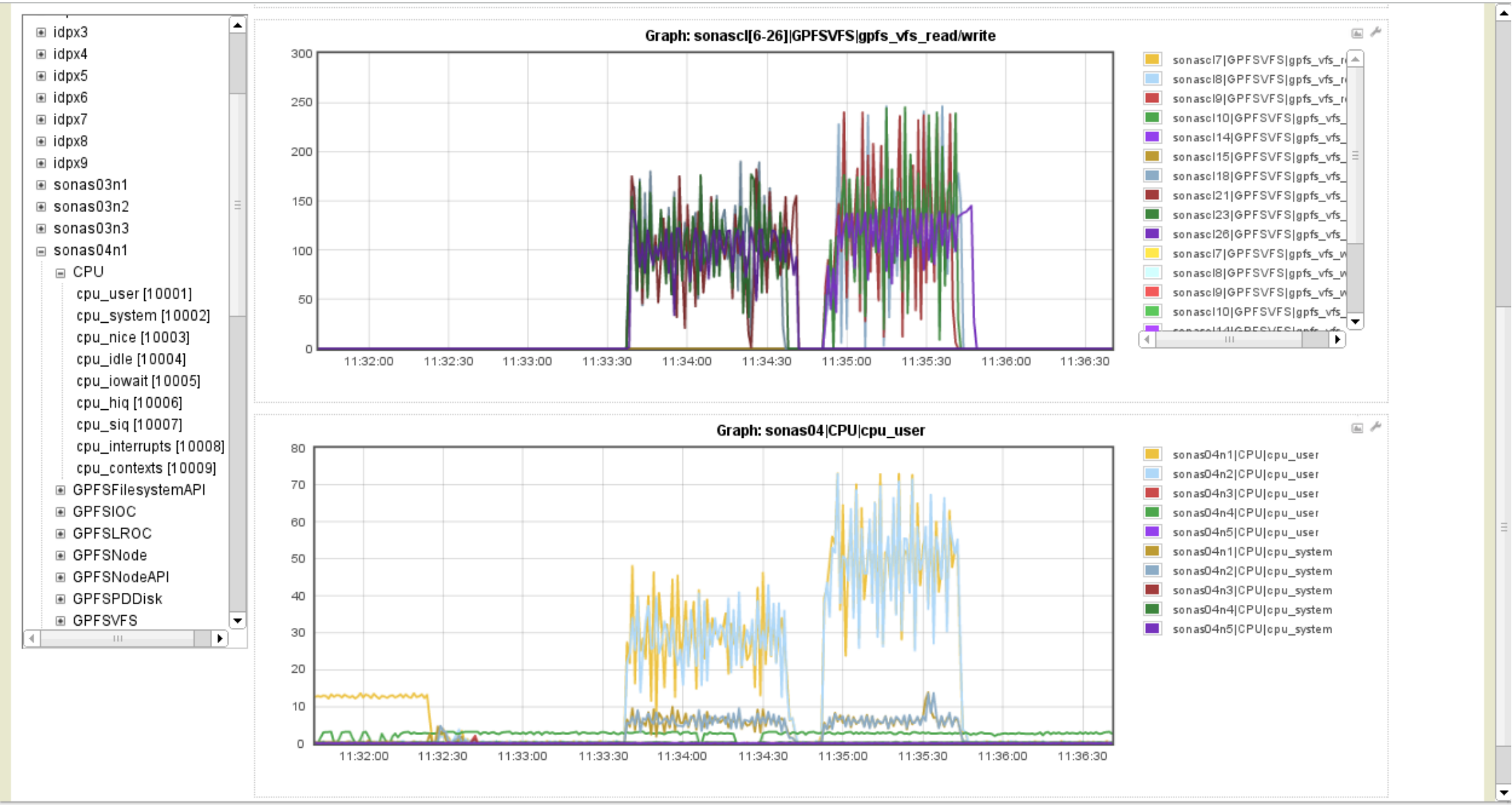
- Initially, with all data coming from the disk storage system, the client reads data from the SAS disks at ~ 5,000 IOPS
- As more data is cached in Flash, client performance increases to 33,000 IOPS while reducing the load on the disk subsystem by more than 95%

How will samba keep up with this trend and close the gap ?

Multi channel
RDMA
Code optimization

and don't forget performance monitoring its not nice to have its a must !!!!

Realtime performance monitor



GPFS Manila Architecture: Multi-Tenant